

# Spark configuration 정리

# Configuration 제어 위치

- spark-env.sh
  - /koshn/clusters/spark/conf 내에 위치
  - Sourced when running various spark program(**per-machine** basis)
- spark-defaults.conf
  - /koshn/clusters/spark/conf 내에 위치
  - Default system properties included when running spark-submit(**per-application** basis)
- spark-submit
  - full\_eps\_1000.sh에 존재 (/home/statistics\_test/full\_eps\_1000.sh)
- SparkConf()
  - NonIPPool.scala에 존재

## Configuration 적용 우선순위

SparkConf() > spark-submit > spark-defaults.conf

# spark-env.sh: Yarn client mode 옵션

```
SPARK_DRIVER_MEMORY=8G  
SPARK_EXECUTOR_INSTANCES=4  
SPARK_EXECUTOR_CORES=2  
SPARK_EXECUTOR_MEMORY=4G  
SPARK_WORKER_INSTANCES=3  
SPARK_WORKER_MEMORY=24G  
SPARK_WORKER_DIR=/kospin/data/tmp  
SPARK_WORKER_OPTS="-Dx=y"  
SPARK_WORKER_CORES=8  
SPARK_MASTER_OPTS="-Dspark.deploy.defaultCore=6"  
SPARK_LOG_DIR=/kospin/data/log  
ulimit -n 131072
```

현재 Standalone mode를 사용하고 있으므로,  
spark-env.sh에 없어도 되는 옵션들

# spark-env.sh : worker 관련 옵션

```
SPARK_DRIVER_MEMORY=8G  
SPARK_EXECUTOR_INSTANCES=4  
SPARK_EXECUTOR_CORES=2  
SPARK_EXECUTOR_MEMORY=4G  
SPARK_WORKER_INSTANCES=3  
SPARK_WORKER_MEMORY=24G  
SPARK_WORKER_DIR=/kospin/data/tmp  
SPARK_WORKER_OPTS="-Dx=y"  
SPARK_WORKER_CORES=8  
SPARK_MASTER_OPTS="-Dspark.deploy.defaultCore=6"  
SPARK_LOG_DIR=/kospin/data/log  
ulimit -n 131072
```

- SPARK\_WORKER\_INSTANCES: 노드 당 worker의 수
- SPARK\_WORKER\_MEMORY: worker가 executor에게 제공해야하는 총 메모리 양
- SPARK\_WORKER\_DIR: worker 프로세스의 작업 디렉터리
- SPARK\_WORKER\_OPTS: worker에 대해서만 config 속성 설정
- SPARK\_WORKER\_CORES: 현재 머신에서 worker 당 사용할 코어의 수

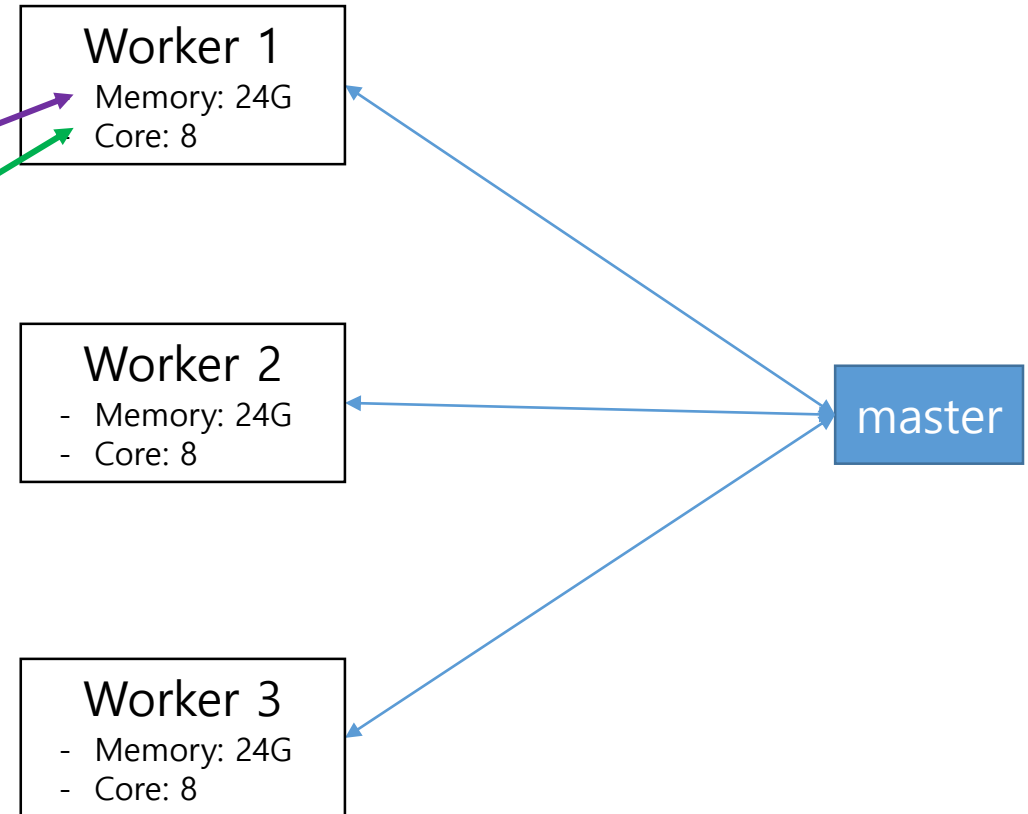
# spark-env.sh: master 관련 옵션

```
SPARK_DRIVER_MEMORY=8G  
SPARK_EXECUTOR_INSTANCES=4  
SPARK_EXECUTOR_CORES=2  
SPARK_EXECUTOR_MEMORY=4G  
SPARK_WORKER_INSTANCES=3  
SPARK_WORKER_MEMORY=24G  
SPARK_WORKER_DIR=/kospin/data/tmp  
SPARK_WORKER_OPTS="-Dx=y"  
SPARK_WORKER_CORES=8  
SPARK_MASTER_OPTS="-Dspark.deploy.defaultCore=6"  
SPARK_LOG_DIR=/kospin/data/log  
ulimit -n 131072
```

- SPARK\_MASTER\_OPTS: master에 대해서만 config 속성 설정
  - spark.deploy.defaultCore 옵션의 경우, spark.cores.max를 설정하지 않은 경우 spark의 standalone 모드에서 애플리케이션에 제공할 기본 코어 수(설정되어 있지 않을 경우, 머신의 모든 코어를 사용)

# Spark 동작 구조 ( spark-env.sh )

```
SPARK_DRIVER_MEMORY=8G  
SPARK_EXECUTOR_INSTANCES=4  
SPARK_EXECUTOR_CORES=2  
SPARK_EXECUTOR_MEMORY=4G  
SPARK_WORKER_INSTANCES=3  
SPARK_WORKER_MEMORY=24G  
SPARK_WORKER_DIR=/kesign/data/tmp  
SPARK_WORKER_OPTS="-Dx=y"  
SPARK_WORKER_CORES=8  
SPARK_MASTER_OPTS="-Dspark.deploy.defaultCore=6"  
SPARK_LOG_DIR=/kesign/data/log  
ulimit -n 131072
```



# spark-defaults.conf : serializer 설정

```
spark.master spark://sniperbd1:7077
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max 1024m
spark.kryoserializer.buffer 64k
spark.shuffle.consolidateFiles true
spark.worker.cleanup.enabled true
spark.driver.allowMultipleContexts true
spark.executor.instances 3
spark.executor.extraJavaOptions -XX: +UseCompressedOops
spark.executor.cores 4
spark.sql.shuffle.partitions 500
spark.jars /koshn/clusters/spark/lib/mysql-connector-java-5.1.31-bin.jar
spark.deploy.defaultCores 6
spark.cores.max 8
```

- spark.serializer: 사용하고자 하는 serializer 설정 (default: java serializer)
- spark.kryoserializer.buffer.max: serializer 최대 버퍼 크기 설정 (직렬화하려는 객체보다 커야하며 2048m 미만이어야 함, 파티션의 모든 데이터를 수용할 수 있을 만큼 커야함)
- spark.kryoserializer.buffer: kryoserializer 버퍼의 초기 크기 설정

# spark-defaults.conf : shuffle 설정

```
spark.master spark://sniperbd1:7077
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max 1024m
spark.kryoserializer.buffer 64k
spark.shuffle consolidateFiles true
spark.worker.cleanup.enabled true
spark.driver.allowMultipleContexts true
spark.executor.instances 17
spark.executor.extraJavaOptions -XX: +UseCompressedOops
spark.executor.cores 4
spark.sql.shuffle.partitions 500
spark.jars /kesign/clusters/spark/lib/mysql-connector-java-5.1.31-bin.jar
spark.deploy.defaultCores 6
spark.cores.max 8
```

셔플 중에 생성된 중간 파일 통합 옵션



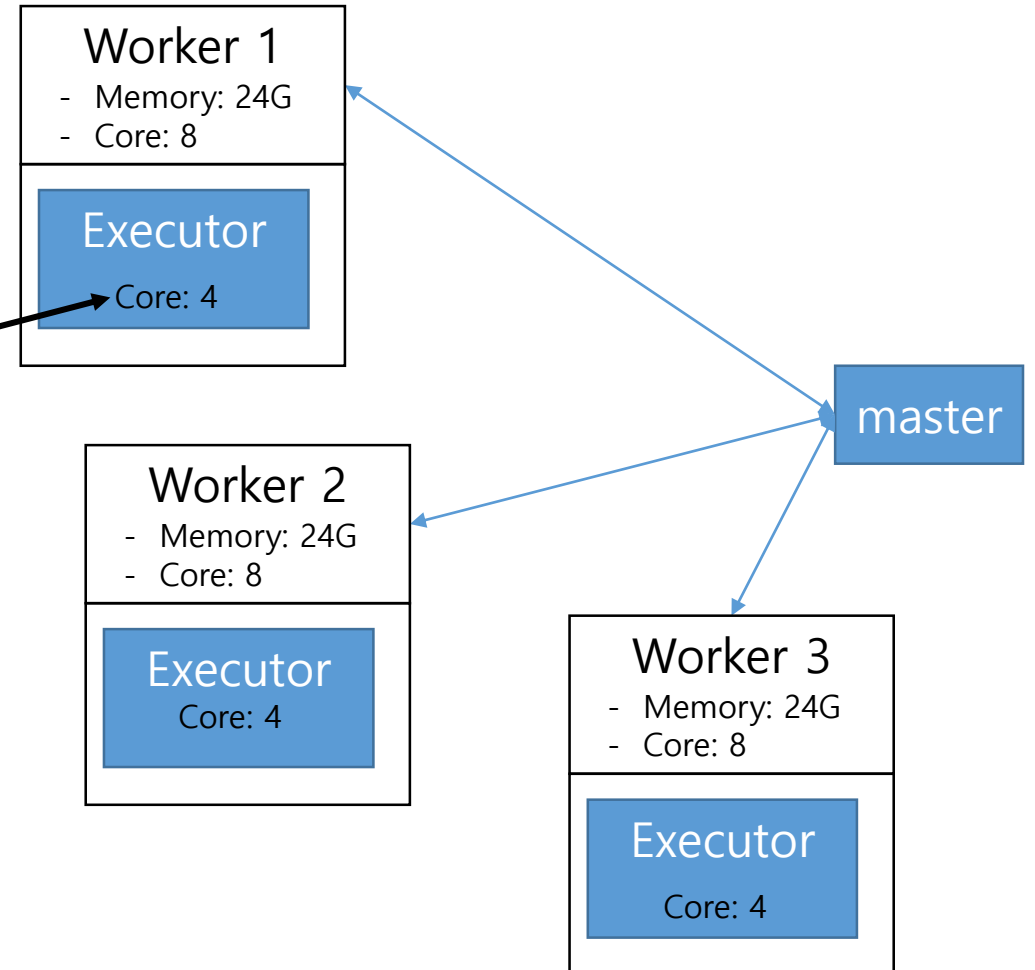
# spark-defaults.conf : executor 설정

```
spark.master spark://sniperbd1:7077
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max 1024m
spark.kryoserializer.buffer 64k
spark.shuffle.consolidateFiles true
spark.worker.cleanup.enabled true
spark.driver.allowMultipleContexts true
spark.executor.instances 17
spark.executor.extraJavaOptions -XX: +UseCompressedOops
spark.executor.cores 4
spark.sql.shuffle.partitions 500
spark.jars /kesign/clusters/spark/lib/mysql-connector-java-5.1.31-bin.jar
spark.deploy.defaultCores 6
spark.cores.max 8
```

- spark.executor.instances: executor의 수 설정(yarn 모드에서만 적용)
- spark.executor.extraJavaOption: executor에 전달할 추가 JVM 옵션 문자열(-XX: +UseCompressedOops 을 통해 memory consumption을 줄일 수 있음)
- spark.executor.cores: 각 executor에서 사용할 코어 수 설정(YARN 모드에서 1, standalone 모드에서 사용 가능한 모든 코어 사용)

# Spark 동작 구조 ( spark-default.conf )

```
spark.master spark://sniperbd1:7077
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max 1024m
spark.kryoserializer.buffer 64k
spark.shuffle.consolidateFiles true
spark.worker.cleanup.enabled true
spark.driver.allowMultipleContexts true
spark.executor.instances 3
spark.executor.extraJavaOptions -XX:+UseCompressedOops
spark.executor.cores 4
spark.sql.shuffle.partitions 500
spark.jars /koshn/clusters/spark/lib/mysql-connector-java-5.1.31-bin.jar
spark.deploy.defaultCores 6
spark.cores.max 8
```



# \*\*Executor

- executor의 총 개수는 다음의 두 가지 조건을 만족하는 최댓값으로 적용됨
  - 1)  $(\text{executor cores}) * (\text{executor의 개수})$  는 total executor cores를 초과할 수 없다. 따라서 executor의 개수는  $(\text{total executor cores}) / (\text{executor cores})$  를 초과할 수 없다.
  - 2) executor는 worker 안에서 memory를 할당 받기 때문에,  $(\text{executor의 memory}) * (\text{executor의 개수})$  가 worker의 memory를 초과할 수 없다. 따라서, executor의 개수는  $(\text{worker memory}) / (\text{executor memory})$  를 초과할 수 없다.
- 단, executor instance에 대한 limit는 dynamic allocation일 때만 finite이고, 그 외의 경우에는 unlimited이기 때문에 executor 개수는 가능한 최대로 적용됨.

# spark-defaults.conf : Scheduling 관련 설정

```
spark.master spark://sniperbd1:7077
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.kryoserializer.buffer.max 1024m
spark.kryoserializer.buffer 64k
spark.shuffle.consolidateFiles true
spark.worker.cleanup.enabled true
spark.driver.allowMultipleContexts true
spark.executor.instances 17
spark.executor.extraJavaOptions -XX: +UseCompressedOops
spark.executor.cores 4
spark.sql.shuffle.partitions 500
spark.jars /kesign/clusters/spark/lib/mysql-connector-java-5.1.31-bin.jar
spark.deploy.defaultCores 6
spark.cores.max 8
```

- spark.deploy.defaultCores:  
(spark.cores.max를 설정하지 않은 경우)  
spark의 standalone 모드에서 애플리케이션에 제공할 기본 코어 수
- spark.cores.max: 각 머신이 아닌 클러스터 전체에서 애플리케이션을 위해 요청할 수 있는 cpu 코어 수

# spark-submit ( full\_eps\_1000.sh )

- --total-executor-cores 4



Spark 작업에 할당 된 모든 실행 프로그램의 총 코어를 나타냄

사용 가능한 코어 수보다 크게 설정해주면 모든 리소스가 사용되므로 코어 수를 제한하는 경우가 아니라면 필요하지 않은 옵션

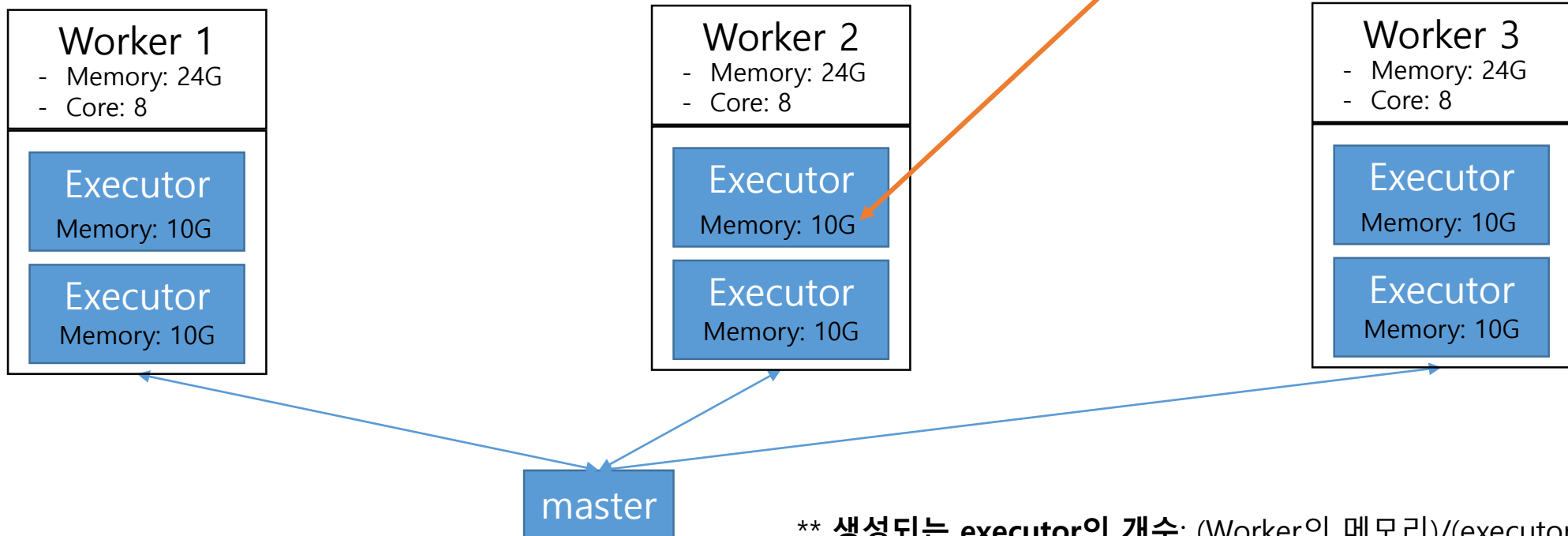
- --executor-memory 10g



익스큐터 프로세서당 사용할 메모리 용량

# Spark 동작 구조 (full\_eps\_1000.sh)

```
/kesign/clusters/spark/bin/spark-submit --class NonIPPool --total-executor-cores 48 --executor-memory 10g $LIB_DIR/ippool_2.12-1.0_eps1000.jar spark://sniperbd1:7077 $Date sniperbd1 1
```



\*\* 생성되는 executor의 개수: (Worker의 메모리)/(executor 1개 메모리)

# SparkConf() ( NonIPPool.scala )

- filtered.cache() (Storage Level : MEMORY\_ONLY)
  - RDD를 여러 번 반복 연산하는 것을 피하려면 데이터 영속화가 필요
  - Storage Level을 변경하여 영속화의 결과를 어떻게 보존할 지 설정 가능
  - \_SER : 원본이 아닌 직렬화 된 형태로 저장

메모리 사용량은 줄일 수 있지만 Serialize와 De-Serialize로 인한 CPU 사용량 증가

```
var filtered = bd1Data.filter(s => (s.rowkey.split("WWW.").length == 2)
&& (s.node != "0") && (Try(s.node.toDouble).isSuccess == true))
filtered.cache()
```

# SparkConf() ( NonIPPool.scala )

영속화 수준	공간 사용	CPU 사용시간	영속화 결과 보존 스토리지	직렬화 여부
MEMORY_ONLY	높음	낮음	메모리	X
MEMORY_ONLY_SER	낮음	높음	메모리	O
DISK_ONLY	낮음	높음	디스크	X
MEMORY_AND_DISK	높음	중간	기본적으로는 메모리에 저장, 용량이 부족해지면 디스크에 보존	O. 디스크에 보존되는 시점에 직렬화
MEMORY_AND_DISK_SER	낮음	높음		O. 디스크에 보존되는 시점, 메모리에 보존되는 시점에 직렬화